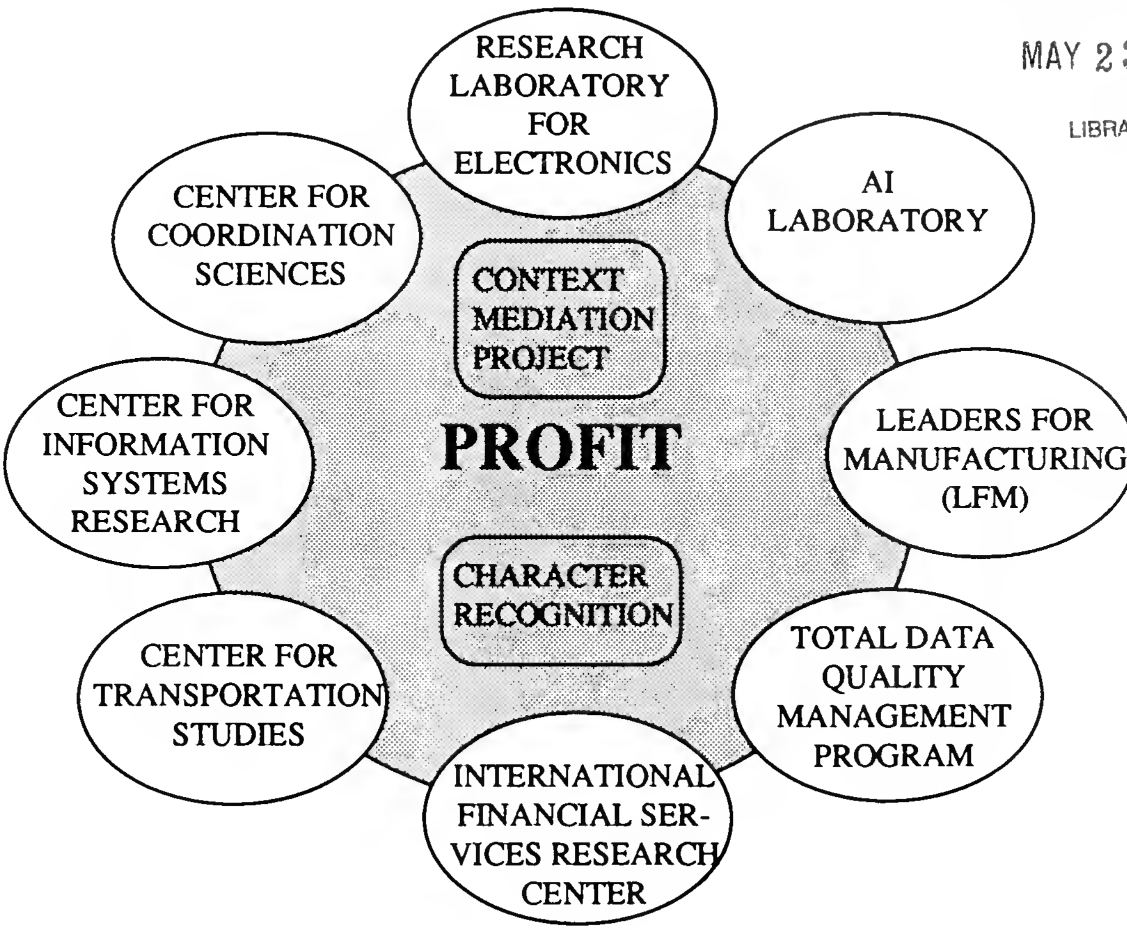
Algorithms for Thinning and Rethickening Binary Digital Pattern

M.V. Nagendraprasad
Patrick Wang
Amar Gupta

WP #3764 1993
PROFIT #93-03

Productivity From Information Technology (PROFIT)

The Productivity From Information Technology (PROFIT) Initiative was established on October 23, 1992 by MIT President Charles Vest and Provost Mark Wrighton "to study the use of information technology in both the private and public sectors and to enhance productivity in areas ranging from finance to transportation, and from manufacturing to telecommunications." At the time of its inception, PROFIT took over the Composite Information Systems Laboratory and Handwritten Character Recognition Laboratory. These two laboratories are now involved in research related to context mediation and imaging respectively.



In addition, PROFIT has undertaken joint efforts with a number of research centers, laboratories, and programs at MIT, and the results of these efforts are documented in Discussion Papers published by PROFIT and/or the collaborating MIT entity.

Correspondence can be addressed to:

The "PROFIT" Initiative
Room E53-310, MIT
50 Memorial Drive
Cambridge, MA 02142-1247
Tel: (617) 253-8584
Fax: (617) 258-7579
E-Mail: profit@mit.edu

EXECUTIVE OVERVIEW

Financial enterprises rely heavily on paper-based documents to conduct various operations; this is true both for external operations involving customers and other financial institutions, as well as internal operations involving various departments.

Researchers at MIT have looked at the possibility of taking information directly from paper documents, especially handwritten documents, to computer-accessible media. Automated reading involves several steps as follows:

(i) Scanning of document;
(ii) Location of area to be "read";
(iii) Decomposing the selected area into separate characters;
(iv) Adjusting size and slant of each character;
(v) Recognizing each character; and
(vi) Testing whether input has been correctly read.

Based on several years of sustained research, the researchers have attained very high "reading" speed and accuracy, even in situations where the quality of the input material is poor. Patent rights for some of the new techniques have been applied for. Sponsor companies are eligible to test the new techniques in their respective environments at no charge.

The work performed so far is described in a number of published paper and working papers. The list of working papers is as follows:

IFSRC # 107-89	Optical Image Scanners and Character Recognition Devices: A Survey and New Taxonomy	Amar Gupta Sanjay Hazarika Maher Kallel Pankaj Srivastava
IFSRC # 123-90R	An Improved Structural Technique for Automated Recognition of Handprinted Symbols Revised October 1990	Patrick S. P. Wang Amar Gupta
IFSRC # 124-90	Integration of Traditional Imaging, Expert Systems, and Neural Network Techniques for Enhanced Recognition of Handwritten Information	Evelyn Roman Amar Gupta John Riordan
IFSRC # 151-91	Handwritten Numeral Recognition Using Dynamic Programming Neural Networks on an Off-Line Basis	Ronjon Nag Alexis Lui Amar Gupta
IFSRC # 162-91R PROFIT 93-03	Algorithms for Thinning and Rethickening Binary Digital Patterns	M. Nagendraprasad Patrick S. Wang Amar Gupta
IFSRC # 173-91	A New Algorithm for Slant Correction of	Vanessa C. Feliberti

	Handwritten Characters	Amar Gupta
IFSRC # 214-92	An Algorithm for Segmenting Handwritten Numeral Strings	Peter L. Sparks M. V. Nagendraprasad Amar Gupta
IFSRC # 215-92	A New Algorithm for Correcting Slant in Handwritten Numerals	M. V. Nagendraprasad Amar Gupta Vanessa Feliberti
IFSRC # 218-92	A System for Automatic Recognition of Totally Unconstrained Handwritten Numerals	M. V. Nagendraprasad
IFSRC # 219-92	A Collection of Papers on Handwritten Numeral Recognition	Amar Gupta
IFSRC # 261-93	An Adaptive Modular Neural Network with Application to Unconstrained Character Recognition	Lik Mui Arun Agarwal P. S. P. Wang
IFSRC # 287-94 PROFIT 93-04	An Integrated Architecture for Recognition of Totally Unconstrained Handwritten Numerals	Amar Gupta M. V. Nagendraprasad A. Liu Amar Gupta S. Ayyadurai
IFSRC # 288-94 PROFIT 93-09	Detection of Courtesy Amount Block on Bank Checks	Arun Agarwal Len M. Granowetter Amar Gupta P. S. P. Wang
IFSRC # 289-94 PROFIT 94-14	A Knowledge Based Segmentation Algorithm For Enhanced Recognition of Handwritten Courtesy Amounts	Karim Hussein Amar Gupta Arun Agarwal Patrick Shen-Pei Wang

The research has been funded by a number of organizations, via the International Financial Services Research Center (IFSRC) and the Productivity from Information Technology (PROFIT) Initiative. Individuals in such sponsor companies should contact their designated contact person at MIT to receive copies of the papers, and the software developed at MIT.

The Principal Investigator for the imaging area is Dr. Amar Gupta, Co-Director, "PROFIT" Initiative, MIT Sloan School of Management, Room E53-311, 77 Massachusetts Avenue, Cambridge, MA 02139, USA; Telephone: (617)253-8906; Fax: (617)258-7579; e-mail: agupta@mit.edu. Your comments and suggestions are encouraged.

DIGITAL SIGNAL PROCESSING 3, 97–102 (1993)

Algorithms for Thinning and Rethickening Binary Digital Patterns

M. V. Nagendraprasad, Patrick S. P. Wang,* and Amar Gupta†

Massachusetts Institute of Technology, Cambridge, Massachusetts 02139

1. INTRODUCTION

Pattern recognition and image processing applications frequently deal with raw inputs that contain lines of different thickness. In some cases, this variation in the thickness is an asset, enabling quicker recognition of the features in the input image. For example, in processing aerial photographs, detection of major landmarks can be aided by the variations in the thickness of the contours. In other cases, the variation can be a liability, and can cause degradation in the accuracy and the speed of recognition. For example, in the case of handwritten characters, the degree of uniformity of the thickness of individual strokes directly impacts the probability of successful recognition, especially if neural network based recognition techniques are employed.

For the latter category of applications, uniform thickness can be attained, prior to recognition stage, by first thinning the input pattern to a thickness of a single pixel and then rethickening it to a constant thickness. The basic structure and the connectivity of the original pattern can be preserved irrespective of the underlying complexity, through the stages of thinning and rethickening.

Digitized bitmap patterns consist of an array of pixels, where each pixel is either 1 ("on" pixel) or 0 ("off" pixel). In thinning, also called skeletonization, the redundant "on" pixels are eliminated from the original pattern to yield its equivalent skeletonized pattern. During the subsequent stage of rethickening, "on" pixels are systematically added to reconstruct an equivalent of the original pattern. Because the thinning process is usually considered more difficult than the rethickening process, the bulk of this paper deals with thinning aspect.

Section 2 deals with basic notation. The thinning stage is discussed in Section 3. Section 4 presents a theoretical proof related to a new and faster thinning algorithm. The rethickening stage is discussed in Section 5. Results are presented in Section 6 and conclusions in Section 7.

* College of Computer Science, Northeastern University.

† This research was funded by the International Financial Services Research Center at MIT's Sloan School of Management. Comments and suggestions should be addressed to the principal investigator for this project: Dr. Amar Gupta, Room E53-311, MIT, Cambridge, MA 02139, USA; telephone (617) 253-8906.

2. BASIC NOTATION

One of the authors [11] has previously presented definitions and notation related to the thinning algorithms presented here. In order to facilitate a direct comparison of the new algorithm with a previous one proposed in [11], the same notation is utilized in this paper.

DEFINITION 1. The neighbors of a pixel, p:$[i, j]$, are identified by the eight directions, $[i-1, j]$, $[i-1, j+1]$, $[i, j+1]$, $[i+1, j+1]$, $[i+1, j]$, $[i+1, j-1]$, $[i, j-1]$, $[i-1, j-1]$. The directions are also assigned a number k taking values from $0, \ldots, 7$ as shown in Fig. 1.

DEFINITION 2. The contour points of a digital pattern are defined as those pixels for which at least one neighbor is off. In Fig. 2, "a," "b," . . . , "k" and some of the pixels 'm' and 'n' are contour points while none of the "1" s is a contour point.

DEFINITION 3. The contour loop is a set of contour points which are connected into a loop. More formally, a set of contour points $c_1, c_2, \ldots, c_n$ (for $(n > 1)$ form a loop iff c_i is a neighbor of c_{i+1} for $1 \leq i < n$ and c_n is a neighbor of c_1. We use $L(1), \ldots, L(m)$ to label the

1051-2004/93 $4.00

p[7]: [i-1, j-1]	p[0]: [i-1, j]	p[1]: [i-1, j+1]
p[6]: [i, j-1]	p: [i, j]	p[2]: [i, j+1]
p[5]: [i+1, j-1]	p[4]: [i+1, j]	p[3]: [i+1, j+1]

FIG. 1. Pixel and its neighbors.

different contour loops of a pattern, where m is the number of contour loops. For example, the digit "3" in Fig. 2 has one contour loop pattern ($m = 1$).

DEFINITION 4. When a contour point u is processed, the next contour point to be processed, v, is called a successor contour point of u and u is called a previous contour point of v. In Fig. 2, if the pixel "d" is processed, then the point "c" is the previous contour point of "d" and "e" is the successor contour point of "d."

DEFINITION 5. Let p be the processing contour point. Let x be the previous contour point of p, and z be the successor contour point of p. Then s will be called a successor function if $z = s(x, p)$ is a function from x according to clockwise order around the neighbors of p to meet the first 1 pixel z. In Fig. 2, if the current processing pixel is the 'f' at the top right corner, and the previous contour point of "f" is "e," then $z = s($"f," "e"$)$ is "h."

3. THINNING ALGORITHM

Thinning algorithms proposed by various researchers [1–11] can be generally grouped into three categories: serial, parallel, and maximum methods. In serial thinning algorithms, the value of a pixel at the nth iteration depends on a set of the pixels for some of which, the result of nth iteration is already known. In contrast, the value of a pixel at the nth iteration of a parallel thinning algorithm depends only on the values of that pixel and its neighbors at the $(n - 1)$th iteration; as such, all the pixels of the digital pattern can be thinned simultaneously, enabling these algorithms to offer faster speeds than the equivalent serial thinning algorithms. The third category of algorithms focuses on maximization of particular functions during the process of skeletonization.

The thinning method for digital patterns proposed in [11] provides an efficient thinning process with respect to parallelism and connectivity. However, the algorithm involves a number of time consuming steps which can be improved upon. As such, a new but functionally equivalent algorithm incorporating these improvements is discussed below.

The bitmap is stored in a matrix Q and the following functions are used by the algorithms.

1. *initial:* a function that:
   (a) computes the contour loop number m;
   (b) computes the first contour point FIRST$[k]$, and its previous point, PREV$[k]$, for each contour loop $L[k]$, $(k = 1, 2, \ldots, m)$; and
   (c) sets the initial value '1' to loop decision variable $h[k]$, where $h[k] = 1$ means that the contour loop points need to be thinned.
2. *loop-end-test* (k): a function that tests whether the kth contour loop is terminated.
3. *contour*(p): a function that tests whether a point is lying on a contour or not. It just looks for a white space ("off" pixel) in the neighborhood of the point.
4. *successor* (x, p): a function that computes the successor point from the previous point x according to clockwise order around the neighbors of p to meet the first "on" pixel.
5. *deletion* (Q, p): a function that deletes a point p in Q (i.e., sets a 1_pixel to 0) if it satisfies the following two conditions:
   (a) $1 < B(p) < 7$; and
   (b) $(A(p) = 1$ or $C(p) = 1)$ and $(p[2] + p[4])*p[0]*p[6] = 0)$ or $(A(p) = 1$ or $C(p) = 1)$ and $(p[0] + p[6])*p[2]*p[4] = 0)$,

where $B(p)$ is the number of neighbors of p which are not equal to zero. $A(p)$ is the number of off-to-on transitions when the neighbors are taking a clockwise walk around p (i.e., along the neighbors of pixel p).

$$C(p) = \begin{cases} 1 & p[k-1] + p[k] + p[k+1] \\ & + p[k+4] = 0 \\ & \text{and } p[k+3] = 1 \text{ and } p[k+5] = 1 \\ & k = 1 \text{ or } k = 3 \\ 0 & \text{otherwise} \end{cases}$$

```
abodefgabodef
g11111111111h
ghijkmnn1111n
        1111 1j
         k1111m
        nn111mm
     nnnn1111mm
     nnnn1111mm
     nnnn1111mm
        nn111mm
         k1111m
        _k1111m
 ghijkmnn1111n
 g11111111111h
abodefgabodef
```

FIG. 2. Digit pattern "3." Note: In the actual data only two types of pixels exist—0 or 1.

The parallel thinning algorithm by Wang and Zhang in [11] performed thinning as follows:

```
Algorithm WZ:
initial; g = 1;
repeat
  Q = Q1; if (g == 1) then g = 0; else g = 1;
  for k = 1 to m do if h[k] = 1 then
    begin p = FIRST[k]; x = PREV[k];
    h[k] = 0;
      repeat
        z = successor (x, p); x = p;
        case g of
          0: if 1 < B(p) < 7 and (A(p) = 1 or
                                  c(p) = 1)
                                  and (p[2] +
                                  p[4])*p[0]*
                                  p[6] = 0
            then begin deletion (Q1, p);
            h[k] = 1 end;
          1: if 1 < B(p) < 7 and (A(p) = 1 or
                                  c(p) = 1)
                                  and(p[0] +
                                  p[6])
                                  *p[2]* p[4] = 0
            then begin deletion (Q1, p);
            h[k] = 1 end;
        end case;
          p = z;
        until loop-end-test (k)
    end
until h[1] + ··· + h[m] = 0.
```

The proposed new parallel algorithm operates as follows:

```
Algorithm NWG:
g = 1;
repeat
  h = 0;
  Q = Q1; if g == 1 then g = 0; else g = 1;
  for each array element p of Q
    begin if contour(p) then
      case g of
        0: if 1 < B(p) < 7 and (A(p) = 1 or
                                c(p) = 1)
                                and (p[2] +
                                p[4])*p[0]*
                                p[6] = 0
          then begin deletion (Q1, p); h[k]
          = 1; h = 1; end;
        1: if 1 < B(p) < 7 and (A(p) = 1 or
                                c(p) = 1)
                                and (p[0] +
                                p[6])*p[2]*
                                p[4] = 0
          then begin deletion (Q1, p); h[k]
          = 1; h = 1; end;
      end case;
    end for;
until h = 0;
```

In each iteration, the new algorithm uses only a subset of the operations involved in the earlier algorithm. During a single pass over the full array, Algorithm NWG uses two operations—*contour*(p) and *deletion* ($Q1$, p), whereas Algorithm WZ used *initial*, *loop-end-test*(k), *successor*(x, p), and *deletion* (Q, p), FIRST[k], PREV[k] and *contour*(p) as a part of initial, FIRST[k] and PREV[k]. Further, the function *contour*(p) is used at least as many times in FIRST[k] as it is in Algorithm NWG; at least one scan is needed to detect the starting points of all the contour loops in the array. The speed of the new algorithm is significantly higher than the earlier one. Also the new algorithm can be programmed more readily.

4. FUNCTIONAL EQUIVALENCE PROOF

This section provides a proof that the new algorithm is operationally equivalent to the earlier algorithm, that identical intermediate outputs occur at the end of each iteration for both cases, and that the final outputs will also be identical.

LEMMA 1. *If the data array at the beginning of ith pass[1] is identical for both the algorithms, and further if Algorithm WZ marks a point p for deletion during the pass in the data array, then Algorithm NWG also marks the same point p for deletion.*

Proof. For Algorithm WZ, to mark point p for deletion, the following conditions must be satisfied:

- point p must be on one of the m contour loops; and
- the case statement must have been satisfied.

But if the point p is a part of any contour loop, then *contour* (p) must be satisfied. If *contour* (p) is satisfied and the case statement is satisfied, then Algorithm NWG will mark the point p for deletion.

LEMMA 2. *If the data array at the beginning of ith pass is identical for both the algorithms, and further if Algorithm NWG marks a point p for deletion during the pass in the data array, then Algorithm WZ also marks the same point p for deletion.*

Proof. For Algorithm NWG, to mark point p for deletion, the following conditions must be satisfied:

[1] Here a pass means the number of times $Q = Q1$ statement is executed. If this algorithm is modified for execution on a parallel computer, then the step involving $Q = Q1$ will not exist. Then the number of passes over the array is the number of times $Q = Q1$ gets executed.

If a point p is marked out for deletion by Algorithm NWG, then the following conditions must be satisfied:

- *contour* (p) must be true; and
- the case statement must have been satisfied.

But if *contour*(p) is true, then p must belong to one of the m contours and hence must be one of the points encountered while Algorithm WZ traverses the contour loops. When this point is encountered and if the case statement is satisfied, then Algorithm WZ will mark the point p for deletion.

LEMMA 3. *Algorithm WZ and Algorithm NWG execute an identical number of passes over the data array.*

Proof. The proof is obvious from the way the two algorithms are constructed. During a pass, Algorithm WZ goes over all the contours in the array once and if there is a deletion during the traversal of any of the

FIG. 3. Examples of digit bitmaps and their skeletonized versions.

contour points, then there is another iteration over the data array. Algorithm NWG goes over all the points in the array and deletes only points that reside on the contour. If there is any deletion during the pass, then it goes through another iteration. Since at the end of any intermediate iteration, the same output is produced by both of them, they will terminate after an identical number of iterations.

THEOREM. *Algorithm NWA and Algorithm WZ are operationally equivalent, that is, given the same input, they produce the same output.*

Proof. Follows from Lemmas 1, 2, and 3 above.

5. RETHICKENING

The skeletonization algorithm above was developed as a part of a larger ongoing project on automated handwritten numerals. The numerals input to a general handwriting recognition system could have been written using various kinds of pens like felt pens, ball pens, or microtipped pens. Each of these writing instruments creates characters of different thickness. In order to reduce these variations, the input pattern is scaled down to a single pixel thickness and then rethickened to a uniform thickness. The algorithm above thins down the numeral pattern to a thickness of one pixel. This "skeletal" bitmap of the numeral is now rethickened to a standard thickness, allowing numerals of varying thickness to be rethickened to a uniform thickness. Even within the bitmap array of a numeral, the thickness of the numeral may be different in different parts. Thinning and rethickening can also smoothen these variations to a large extent.

The algorithm for rethickening looks at a pre-specified neighborhood of each 1_pixel and fills this neighborhood with 1's.

6. RESULTS

The National Institute of Standards and Technology (NIST) Handprinted Character Database was utilized as the basis for providing standardized inputs. The raw data in a bitmap form, with an average size of about 50×350, were normalized to a 16×150 array. In the latter array, the data were typically 4 to 6 pixels in thickness. Figure 3 shows a sample set of numerals before skeletonization and the corresponding skeletonized numerals obtained as the output of the proposed algorithm. Figure 4 shows a few samples of digits which were skeletonized and then rethickened. Note that the digits in the first column, which were the original digits, are of varying thickness. Certain numerals, like '5,' are extremely narrow, whereas certain others, like '1,' are of bigger thickness. Even within the bitmap of some numerals, like "7," there are variations in thickness. The third column represents the rethickened digits which are distinguished by the uniformity in their thickness not only within a numeral bitmap but also within a set of numeral bitmaps.

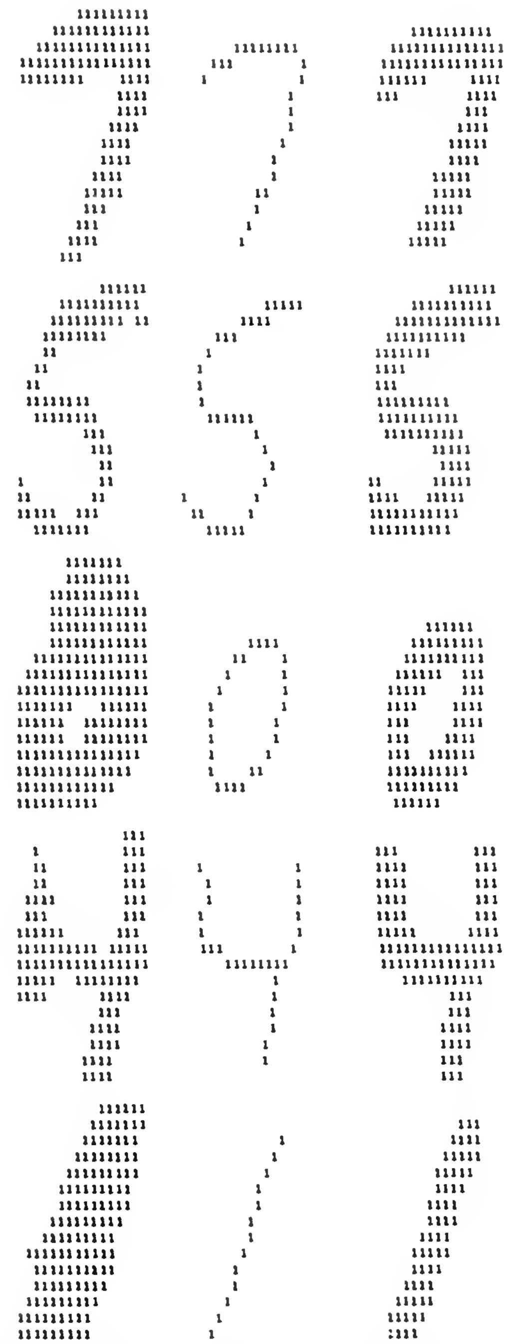


FIG. 4. Examples of numerals, and their skeletonized and rethickened versions.

7. CONCLUSION

Apart from presenting the motivation for thinning and rethickening, this paper has discussed algorithms for performing these tasks efficiently. The new thinning algorithm has been shown to be faster, but functionally identical to, one of the fastest thinning algorithms in existence. By using the algorithms presented in this paper, raw inputs with underlying deficiencies in terms of unequal thickness can be normalized to yield higher overall speed and accuracy.

REFERENCES

1. Adairid, M. R., Martin, G., and Suen, C. Y., Tracing centerlines of digital patterns using Maximal-Square and Maximal-Circle Algorithms. Proc. of Vision Interface' 90, Halifax, Canada, 1990, pp. 67–79.
2. Arcelli, C., Cordella, L. P., and Levialdi, S., From local maxima to connected skeletons. *IEEE Trans. Pattern Anal. Mach. Intell.* 3 (1981), 134–143.
3. Arumugam, A., Radhakrishnan, T., and Suen, C. Y., A thinning algorithm based on the force charged particle and its experimental comparison. Personal communication.
4. Davies, E. R., Plummer, A. P. N., Thinning algorithms: A critique and a new methodology. *Pattern Recognition* **14**, 1 (1981), 53–63.
5. Lu, H. E., and Wang, P. S. P., A comment on "A fast parallel algorithm for thinning digital patterns." *Commun. ACM* **29** (1986), 239–242.
6. Naccache, N. J., and Shinghal, R., An investigation into skeletonization approach of Hilditch. *Pattern Recognition* **17**, 3 (1984), 274–284.
7. Pavalidis, T., A flexible parallel thinning algorithm. In Proceedings of IEEE Computer Society Conference on Pattern Recognition and Image Processing, 1981, pp. 162–167.
8. Sossa, J. H., An improved parallel algorithm for thinning digital patterns. Pattern Recognition Lett. (1989), 77–80.
9. Stefanelli, R., and Rosenfeld, A., Some parallel thinning algorithms for digital pictures. *J. ACM* **18** (1971), 255–264.
10. Tamura, H., A comparison of line thinning algorithms from digital geometry viewpoint. In the Proceedings of 4th International Conference on Pattern Recognition, 1978, pp. 715–719.
11. Wang, P. S. P., and Zhang, Y. Y., A fast and flexible thinning algorithm. *IEEE Trans. Comput.* **38**, 5 (1989).

M. V. NAGENDRAPRASAD obtained his B.Tech from Indian Institute of Technology, Madras, and M.S. degrees from Indian Institute of Technology, Madras, and Massachusetts Institute of Technology. He is currently a student at the University of Massachusetts, Amherst, working towards a doctorate in Distributed Artificial Intelligence. His interests include artificial intelligence, connectionist systems, machine learning, and cognitive science.

PATRICK S. P. WANG is a Professor of Computer Science at Northeastern University. He received his Ph.D. degree in computer science from Oregon State University, his M.S.I.C.S. degree from Georgia Institue of Technology, his M.S.E.E. degree from National Taiwan University, and his B.S.E.E. degree from National Chiao Tung University.

Before joining Northeastern University, he had been with WANG Laboratories, GTE Laboratories, Boston University, and the University of Oregon as software engineering specialist, senior member of technical staff, adjunct Associate Professor, and Assistant Professor, respectively. From 1989 to 1990, he was a visiting scientist at the MIT Artificial Intelligence Laboratory. He is now also teaching part-time at Harvard University, and is a research consultant at the MIT Sloan School of Management.

Professor Wang has published six books and over 70 technical papers in imaging technology, pattern recognition, and artificial intelligence. One of his most recent inventions concerning pattern recognition systems has been granted a patent by the U.S. Patent Bureau. He has attended and organized numerous international conferences for IEEE, ACM, ICS, AAAI, CLCS, and the International Association for Pattern Recognition. He is also a reviewer for grants proposals of the NSF Intelligent Systems Division, and for several computer journals including *IEEE-PAMI, Computer Vision, Graphics, and Image Processing, Information Sciences*, and *Information Processing Letters*, and an editor-in-charge of the *International Journal of Pattern Recognition and Artificial Intelligence.*

Prof. Wang is a senior member of IEEE, and also a member of ACM, AAAI, and the Pattern Recognition Society. Dr. Wang is chairman of IAPR-SSPR (1992–1994) and has been a governing board member and the newsletter editor of the Chinese Language Computer Society (CLCS), Washington, DC (1983–1989). His research interests include software engineering, programming languages, pattern recognition, artificial intelligence, image technology, and automation.

Professor Wang has also written several articles on the operas of Verdi, Puccini, and Wagner, and on Beethoven's, Mozart's and Tsaikovsky's symphonies.

AMAR GUPTA (SM'85) is Senior Research Scientist at the Sloan School of Management, Massachusetts Institute of Technology, Cambridge. He holds a bachelor's degree in electrical engineering from the Indian Institute of Technology, Kanpur, a master's degree in management from the Massachusetts Institute of Technology, and a doctorate in computer technology from the Indian Institute of Technology, New Delhi. He conducted the research for his doctorate dissertation at three prestigious universities in India, England, and the United States.

Dr. Gupta has been involved in research, management, and publishing activities since 1974. Since joining MIT in 1979, he has been active in the areas of multiprocessor architectures, performance measurement, distributed homogeneous and heterogeneous data bases, expert systems, personal computers, and transfer of information technology. He serves as a consultant to a number of corporations, government agencies, and international bodies on various aspects of computer technology, and is an active technical advisor to several international organizations and committees.

Dr. Gupta is the Chairman of the Technical Committee for Microprocessor Applications of the IEEE Industrial Electronics Society, and assistant chairman for several annual IECON conferences. He was also involved with the Very Large Data Base Conference (VLDB '87), held in England in 1987. He received the Rotary Fellowship for International Understanding in 1979 and the Brooks Prize (Honorable Mention) in 1980).

He has written more than 50 technical articles and papers, and produced seven books.

Dr. Gupta is a citizen of the United States.